Alarms Task

Local Station Software Module ALARMS

R. Goodwin

Aug 9, 1989

Function

The Alarms Task performs the alarm scanning of analog channel readings and binary status bits every 15 Hz cycle. If a channel reading is out-of-tolerance, or if a status bit value has the wrong state, its alarm state is "bad." Changes in alarm state are reported by messages sent to an alarm destination node. At the conclusion of the alarm scan, if any channel or bit is in the "bad" state that has the inhibit bit set in its associated alarm flags, an inhibit control line is asserted. This may be used to inhibit beam when key devices are down.

Task events

Event #0 is used to signal that the alarm scan is to be performed. This event is sent by the interrupt routine that runs in response to the 15 Hz interrupt. (The external 15 Hz trigger pulse plugs into a Lemo connector on the Crate Utility Board.) This alarm scan is scheduled to run each cycle after new data has been read and network data requests have been fulfilled.

Event #3 is used to signal invocation of the closed loop code. It is sent by the alarm scan code at 15 Hz; hence, closed loop code runs after the alarm scan finishes. Closed loops can be run at other times that this event is sent to the alarms task. The operation of a closed loop may require this if, say, a closed loop starts an action that results in an interrupt that requires post-processing to be done by the closed loop logic. *Closed loop logic is not used in the Loma Linda system*. The `CLOOPDMY` stub module satisfies the linker.

During the alarm scan, if any alarm messages were queued to the network, event #3 is sent to the Update Task to flush the queued messages to the network.

Option switches

Some of the option switches on the Crate Utility board's front panel influence the handling of alarm messages. The most significant of these is option switch bit#1, the alarm inhibit switch. When it is on, the alarm scan is performed as usual, but alarm messages are *not sent to the network*. They can be displayed locally, however. The inhibit control line is not asserted while this switch is set, in order to prevent having the inhibit line set while sending no network message to say why it is set.

Local display of the alarm messages is handled by the QMonitor Task, as it is the one which "cleans up" after messages that are sent to the network. All alarm messages that result from the alarm scan are queued through `OUTPQ`. But when the alarm inhibit option switch is set, a "used" bit is set that prevents actually sending the message to the network. But QMonitor sees the message as it processes the `OUTPQ` entries and can process such messages also. To enable the local display of alarm messages, set option switch #2 to enable display of the messages on the bottom line of the small screen display, and/or set option switch #3 to enable output of encoded alarm messages to the local serial port. Also, option switch #4 must be set to include analog alarm messages in either type of local display. See "Local Console Alarms Display" document for more information about the "bottom line" alarm message display.

Alarms Task initialization

Various pointers are pre-computed for more efficient processing during the alarm scans.

scan.

Alarm scan

There are two special Bits of status that are used by the alarms task. They are Bits $A0 and $A1 in every node. At the beginning of an alarm scan, each of these special Bits is checked for special processing.

Bit $A0 is set by a host to clear the analog and binary trip counts for all channels and Bits in a node. (A trip count is the count of the number of times a channel or Bit has made an alarm state transition from good to bad.) Typically, a setting command is sent to all nodes via a broadcast setting after producing a report of all the trip counts of interest in every node. The bit is self clearing.

Bit $A1 is set by a host to "reset" the alarms in a node. The bit is set upon system reset by Alarms Task initialization. When the bit is found set, the good/bad status bit in the alarm flags is cleared for all channels and Bits. The result of this is that any channel or bit in alarm will, on the next alarm scan, cause an alarm message to be generated. The bit is self clearing.

Alarm Flags

The format of the alarm flags byte used for either channels or Bits is:

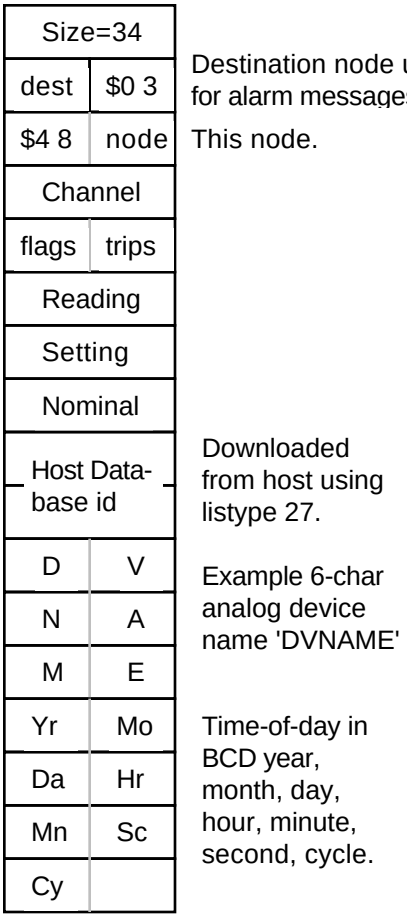
Bit#	Meaning
7	Active. "1": this channel or bit should be scanned.
6	Nominal state. Used only for the binary Bit case.
5	Inhibit. "1": If this channel or bit is bad, assert inhibit control line.
4	Two times. "1": Channel or Bit must be in a new alarm state on two successive scans in order to be recognized as a changed state.
3	Beam only. "1": Only scan this channel or bit on Beam cycles.
2	Not used.
1	Two times counter.
0	Good/bad. "1": bad.

The first 5 bits are control flag bits. The last two are status flag bits.
The inhibit control line is Bit $90. It is assumed memory-mapped. A one means inhibit.
Beam cycles are signaled by the Beam status Bit $9F. A zero means a beam cycle.

Analog alarm scan

For each channel that has the Active bit set, the Beam flag bit is checked. If it is set, but the current cycle is not a beam cycle, then the channel is skipped. The difference between the current reading and the nominal value is compared in absolute value against the tolerance value. If the reading is outside of tolerance, the channel is judged to be bad, and a message is sent if the good/bad bit is zero. (The trip count is also incremented, latching if it reaches 255.) If it is within tolerance, it is judged to be good, and a message is sent if the good/bad bit is a one. The "Two times" bit being set filters this judgment such that the same alarm state must be found on two successive cycles before action is taken. The "two times counter" provides the memory for this determination. The tolerance is a signed value that must be positive; hence, a channel that is 10 volts away from the nominal value is considered to be out-of-tolerance. This restriction could be removed if the tolerance were considered to be unsigned.

Analog alarm message

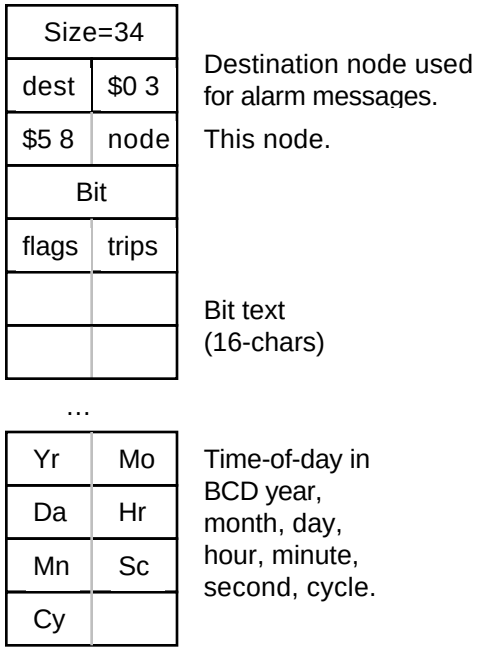


Binary alarm scan

For each Bit that has the Active bit set, the Beam flag bit is checked. If it is set, but the current cycle is not a beam cycle, then the Bit is skipped. If the current Bit reading differs from the Bit's Nominal flag bit state, and the good/bad bit in the alarm flags byte is zero, then send an alarm message to the destination node for alarms. (The trip count is also incremented, latching if it reaches 255.) If the reading matches the Nominal state, and the good/bad bit is a one, then send an alarm message. As in the case of the analog scan, the "Two times" flag bit modifies this determination.

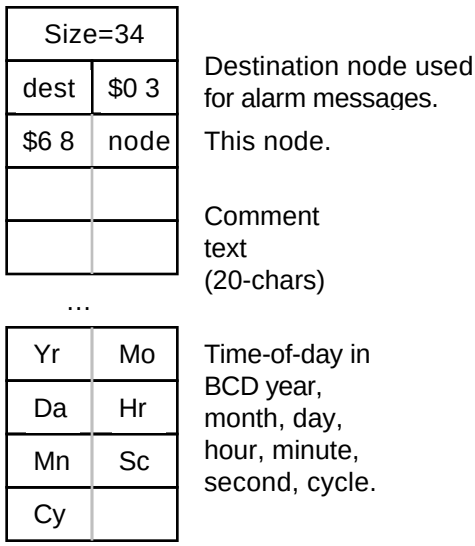
Binary alarm message

The format of an binary alarm message that is queued to the network is as follows:



Comment alarms

A pure text alarm message can also be generated. Currently, it is only used for a message that announces that a VME system has just reset. The Comment routine is included in the Alarms Task module to build such a text message for the network. The format of the network message is as follows:



Timing

The alarm flags byte is scanned sequentially for every channel and bit known to the system. Only channels or bits with the Active bit set are checked for alarm conditions. The three instruction loop that does this is optimized for efficiency. The overhead to scan 256 channels *or* 256 bits *without* the active bit set requires about 250 µ sec on a 133A cpu board. The total Alarm Task time on such an empty system is 0.7 msec. For each active channel or bit that must be scanned, additional time is taken. Rough measurements indicate that it takes 5 µ sec to scan a

An improvement that could be made in the alarm scan logic would be to find a way to check only the channels or bits that need to be scanned. A way to handle *binary* scanning more efficiently would be to do logical operations on 16 binary status bits at a time, rather than checking one bit at a time as is done here. But one must weigh the advantage of such changes that would reduce the scan time against the effort required to make the changes.